CSAIL

massachusetts institute of technology — computer science and artificial intelligence laboratory

Contextual Models for Object Detection Using Boosted Random Fields

Antonio Torralba, Kevin P. Murphy and William T. Freeman

AI Memo 2004-013 June 2004

Abstract

We seek to both detect and segment objects in images. To exploit both local image data as well as contextual information, we introduce Boosted Random Fields (BRFs), which uses Boosting to learn the graph structure and local evidence of a conditional random field (CRF). The graph structure is learned by assembling graph fragments in an additive model. The connections between individual pixels are not very informative, but by using dense graphs, we can pool information from large regions of the image; dense models also support efficient inference. We show how contextual information from other objects can improve detection performance, both in terms of accuracy and speed, by using a computational cascade. We apply our system to detect stuff and things in office and street scenes. [1].

[1] This work was sponsored in part by the Nippon Telegraph and Telephone Corporation as part of the NTT/MIT Collaboration Agreement, and by DARPA contract DABT63-99-1-0012.

1 Introduction

Our long-term goal is to build a vision system that can examine an image and describe what objects are in it, and where. In many images, such as Fig. 6(a), objects of interest, such as the keyboard or mouse, are so small that they are impossible to detect just by using local features. Seeing a blob next to a keyboard, humans can infer it is likely to be a mouse; we want to give a computer the same abilities. Murphy et al used global context to help object recognition [8], but did not model relationships between objects; Fink and Perona [3] exploited local dependencies, but without a guiding probabilistic framework.

In this paper, we exploit contextual correlations between the object classes by introducing Boosted Random Fields (BRFs). Boosted random fields build on both boosting [4, 9] and conditional random fields (CRFs) [7, 6, 5]. Boosting is a simple way of sequentially constructing "strong" classifiers from "weak" components, and has been used for single-class object detection with great success [11].

Conditional random fields (CRFs) are a natural way to model correlation between labels (the outputs of classifiers), given an image as input. The main problem with CRFs is how to learn the correlation (graph) structure of the model. A 4-nearest neighbor grid structure, successful in low-level vision, will fail in capturing important long distance dependencies between whole regions and across classes. In BRFs, we learn the graph structure by using boosting to select from a dictionary of connectivity templates (derived from labeled segmentations), which are combined together in an additive model. We also use boosting to learn the local evidence potentials of the model. We interleave the learning steps with approximate inference, based on belief propagation.

In addition to recognizing things, such as cars and people, we are also interested in recognizing spatially extended "stuff" [1], such as roads and buildings. The traditional sliding window approach to object detection does not work well for detecting "stuff". Instead, we combine object detection and image segmentation c.f., [2]. The desired output will be a coarse segmentation of the image, where some of the regions have semantically meaningful labels attached (see Fig. 6(b)). We will not rely on a bottom-up image segmentation algorithm, which can be fragile without top-down guidance.

2 Learning local evidence potentials on a fixed graph

A conditional random field (CRF) is a distribution of the form

$$P(S|x) = \frac{1}{Z} \prod_i \phi_i(S_i) \prod_{j \in N_i} \psi_{i,j}(S_i, S_j)$$

where x is the input (e.g., image) and N_i are the neighbors of node i. We have assumed pairwise potentials for notational simplicity. Suppose the ψ compatibility potentials are known. Learning the local evidence potentials, ϕ_i, is still a hard problem, even if all the S_i's are known during training, because computing the partition function Z is intractable for most graphs (with the notable exception of chains and trees).[1]

For BRFs, we propose the following simple approximation: use belief propagation (BP) to estimate the marginals, $P(S_i|x)$, and then use boosting to maximize the likelihood of each node's training data wrt ϕ_i. Since the potentials affect the results of BP and vice versa, we suggest the following iterative scheme: perform one round of boosting (add one weak learner to each ϕ_i), then perform one round of BP (send messages to all neighbors), and

[1] In a generative model, such as a Markov random field, we usually define $\phi_i(S_i) = p(x_i|S_i)$, so the local compatibilities are uncoupled from ψ_{ij}, and hence are easy to learn. This is not the case with a CRF.

```
1. Input: a set of labeled pairs $\{x_{i,m}; S_{i,m}\}$, neighbors $\{N_i\}$, pairwise compatibilities $\psi_{ij}$
   and parameters $T, N_{boost}, N_{BP}$. Output: Local evidence functions $f_i^t(x)$.
2. Initialize $F_m^{t=0} = 0$, $w_{i,m}^{t=0} = 1$, $M_i^{t=0}(\pm 1) = 1$, and $\mu_{i\to j}^{t=0}(\pm 1) = 1$.
3. For t=1..T,
   (a) Iterate $N_{boost}$ times
       i. Solve the weighted least squares problem in Eq. 5 wrt $f_i^t$.
       ii. Update local potentials, with $F_i^t = F_i^{t-1} + f_i^t$
       iii. Update the beliefs using Eq. 3.
       iv. Update the weights: $w_{i,m}^{t+1} = b_{i,m}^t(+1)\, b_{i,m}^t(-1)$
   (b) Iterate $N_{BP}$ times
       i. Update messages using Eq. 2.
       ii. Update the beliefs using Eq. 3.
```

Figure 1: Learning the local evidence potentials of a CRF using boosting.

repeat. If the potentials are fixed, this is just regular BP; if the messages are uninformative (e.g., a disconnected graph), this is just regular boosting.

In more detail, the algorithm is as follows. At iteration t, the goal is to minimize the negative log-likelihood of the training data. As in [10], we consider the per-label loss (i.e., we use marginal probabilities), as opposed to requiring that the joint labeling be correct (as in Viterbi decoding). Hence the cost function to be minimized is

$$J^t = \prod_i J_i^t = -\prod_m \prod_i b_{i,m}^t(S_{i,m}) = -\prod_m \prod_i b_{i,m}^t(+1)^{S_{i,m}^*} b_{i,m}^t(-1)^{1-S_{i,m}^*} \qquad (1)$$

where $b_{i,m}^t \approx P(S_i|x_m, t)$ is the belief at node i in training case m given the evidence that has reached it after t iterations. We consider binary states $S_{i,m} \in \{-1, +1\}$, and $S_{i,m}^* = (S_{i,m} + 1)/2$, although the approach could be generalized to higher state dimensions.

The belief at node i is given by the following (dropping the dependence on case m) $b_i^t(\pm 1) \propto \phi_i^t(\pm 1)\, M_i^t(\pm 1)$ where M_i^t is the product of all the messages coming into i from all its neighbors at time t and where the message that k sends to i is given by

$$M_i^{t+1}(\pm 1) = \prod_{k \in N_i} \mu_{k\to i}^{t+1}(\pm 1) \qquad \mu_{k\to i}^{t+1}(\pm 1) = \sum_{s_k \in \{-1,+1\}} \psi_{k,i}(s_k, \pm 1)\, \frac{b_k^t(s_k)}{\mu_{i\to k}^t(s_k)} \qquad (2)$$

where $\psi_{k,i}$ is the compatility between nodes k and i. If we assume that the local potentials have the form $\phi_i^t(s_i) = [e^{F_i^t/2}; e^{-F_i^t/2}]$, where F_i^t is some function of the input data, then it is simple to show that

$$b_i^t(+1) = \sigma(F_i^t + G_i^t), \qquad G_i^t = \log\, M_i^t(+1) - \log\, M_i^t(-1) \qquad (3)$$

where $\sigma(u) = 1/(1 + e^{-u})$ is the sigmoid function. Hence each term in Eq. 1 simplifies to a cost function similar to that used in boosting:

$$\log J_i^t = \sum_m \log\left(1 + e^{-S_{i,m}(F_{i,m}^t + G_{i,m}^t)}\right) \qquad (4)$$

Defining $F_i^t(x_{i,m}) = F_i^{t-1}(x_{i,m}) + f_i^t(x_{i,m})$ as an additive model, where $x_{i,m}$ are the features of training sample m at node i, we can learn this function in a stagewise fashion by optimizing the second order Taylor expansion of Eq. 4 wrt f_i^t, as in logitBoost [4]:

$$\arg\min_{f_i^t} \log J_i^t \simeq \arg\min_{f_i^t} \sum_m w_{i,m}^t (Y_{i,m}^t - f_i^t(x_{i,m}))^2 \qquad (5)$$

```
1. Input: a set of labeled pairs $\{x_{i,m}; S_{i,m}\}$, bound $T$
   Output: Local evidence functions $f^t_i(x)$ and message update functions $g^t_i(b_{N_i})$.
2. Initialize: $b^{t=0}_{i,m} = 0$; $F^{t=0}_{i,m} = 0$; $G^{t=0}_{i,m} = 0$
3. For t=1..T.
   (a) Fit local potential $f_i(x_{i,m})$ by weighted LS to $Y^t_{i,m} = S_{i,m}(1 + e^{-S_{i,m}(F^t_i + G^t_{i,m})})$.
   (b) Fit compatibilities $g^t_i(b^{t-1}_{N_i,m})$ to $Y^t_{i,m}$ by weighted LS.
   (c) Compute local potential $F^t_{i,m} = F^{t-1}_{i,m} + f^t_i(x_{i,m})$
   (d) Compute compatibilities $G^t_{i,m} = \sum_{n=1}^{t} g^n_i(b^{t-1}_{N_i,m})$
   (e) Update the beliefs $b^t_{i,m} = \sigma(F^t_{i,m} + G^t_{i,m})$
   (f) Update weights $w^{t+1}_{i,m} = b^t_{i,m}(-1)\, b^t_{i,m}(+1)$
```

Figure 2: BRF training algorithm.

where $Y^t_{i,m} = S_{i,m}(1+e^{-S_{i,m}(F^t_i+G^t_{i,m})})$. In the case that the weak learner is a "regression stump", $f_i(x) = ah(x)+b$, we can find the optimal a, b by solving a weighted least squares problem, with weights $w^t_{i,m} = b^t_i(-1)\, b^t_i(+1)$; we can find the best basis function h by searching over all elements of a dictionary. Once we have updated F^t_i, and hence ϕ_i, we can compute the new beliefs, b^{t+1}_i, and repeat. See Fig. 1 for the pseudo-code. As noted above, standard boosting and standard belief propagation are just special cases of the BRF algorithm, obtained by setting $N_{BP} = 0$ and $N_{boost} = 0$ respectively. (In the latter case, the local evidence ϕ^t_i must be provided as input.) By combining the two algorithms, we can learn to classify using both local evidence and information from other variables.

3 Learning potentials between nodes and graph structure

In this section, we discuss how to learn the compatibility functions ψ_{ij}, and hence the structure of the graph. We assume that the graph is very densely connected so that the information that one single node sends to another is so small that we can make the approximation $\mu^{t+1}_{k\to i}(+1)/\mu^{t+1}_{k\to i}(-1) \simeq 1$. (This is a reasonable approximation in the case of images, where each node represents a single pixel; only when the influence of many pixels is taken into account will the messages become informative.) Hence

$$G^{t+1}_i = \log \frac{M^{t+1}_i(+1)}{M^{t+1}_i(-1)} = \tag{6}$$

$$\sum_k \log \frac{\sum_{s_k \in [-1,+1]} \psi_{k,i}(s_k, +1)\, \frac{b^t_{k,m}(s_k)}{\mu^t_{i\to k}(s_k)}}{\sum_{s_k \in [-1,+1]} \psi_{k,i}(s_k, -1)\, \frac{b^t_{k,m}(s_k)}{\mu^t_{i\to k}(s_k)}} \simeq \tag{7}$$

$$\sum_k \log \frac{\sum_{s_k \in [-1,+1]} \psi_{k,i}(s_k, +1)\, b^t_{k,m}(s_k)}{\sum_{s_k \in [-1,+1]} \psi_{k,i}(s_k, -1)\, b^t_{k,m}(s_k)} \tag{8}$$

With this simplification, G^{t+1}_i is now a non-linear function of the beliefs $G^{t+1}_i(\vec{b}^t_m)$ at iteration t. Instead of learning the compatibility functions ψ_{ij}, we propose to learn directly the function G^{t+1}_i. We propose to use an additive model for G^{t+1}_i as we did for learning F.

$$G^{t+1}_{i,m} = \sum_{n=1}^{t} g^n_i(\vec{b}^t_m)$$

```
1. Input: a set of inputs $\{x_{i,m}\}$ and functions $f_i^t$, $g_i^t$
   Output: Set of beliefs $b_{i,m}$ and MAP estimates $S_{i,m}$.
2. Initialize: $b_{i,m}^{t=0} = 0$; $F_{i,m}^{t=0} = 0$; $G_{i,m}^{t=0} = 0$
3. From $t = 1$ to $T$, repeat
   (a) Update local evidences $F_{i,m}^t = F_{i,m}^{t-1} + f_i^t(x_{i,m})$
   (b) Update compatibilities $G_{i,m}^t = \sum_{n=1}^t g_i^n(b_{N_i,m}^{t-1})$
   (c) Compute current beliefs $b_{i,m}^t = \sigma(F_{i,m}^t + G_{i,m}^t)$
4. Output classification is $S_{i,m} = \delta\left(b_{i,m}^t > 0.5\right)$
```

Figure 3: BRF run-time inference algorithm.

The weak learners $g_i^n(\vec{b}_m^t)$ can be regression stumps with the form:

$$g_i^n(\vec{b}_m^t) = a\delta(\vec{w} \cdot \vec{b}_m^t > \theta) + b$$

where a, b, θ are the parameters of the regression stump, and $\vec{w}$ is a set of weights selected from a dictionary.

In the case of a graph with weak and almost symmetrical connections (which holds if $\psi(s_1, s_2) \approx 1$, for all (s_1, s_2), which implies the messages are not very informative) we can further simplify the function G_i^{t+1} by approximating it as a linear function of the beliefs:

$$G_{i,m}^{t+1} = \sum_{k \in N_i} \alpha_{k,i}\, b_{k,m}^t(+1) + \beta_{k,i} \tag{9}$$

This step reduces the computational cost. The weak learners $g_i^n(\vec{b}_m^t)$ will also be linear functions.

Hence the belief update simplifies to $b_{i,m}^{t+1}(+1) = \sigma(\vec{\alpha}_i \cdot \vec{b}_m^t + \beta_i + F_{i,m}^t)$, which is similar to the mean-field update equations. The neighborhood N_i over which we sum incoming messages is determined by the graph structure; and the values, $\alpha_{k,i}$ and $\beta_{k,i}$ are determined by the pairwise compatibilities. Instead of trying to learn the compatibilities $\psi_{i,j}$, we propose to learn $\alpha_{k,i}$ and $\beta_{k,i}$ directly. Each weak learner g_i^n will compute a weighted combination of the beliefs of the some subset of the nodes; this subset may change from iteration to iteration, and can be quite large. At iteration t, we choose the weak learner g_i^t so as to minimize

$$\log J_i^t(b^{t-1}) = -\sum_m \log\left(1 + e^{-S_{i,m}(F_{i,m}^t + g_i^t(b_m^{t-1}) + \sum_{n=1}^{t-1} g_i^n(b_m^{t-1}))}\right)$$

which reduces to a weighted least squares problem similar to Eq. 5.

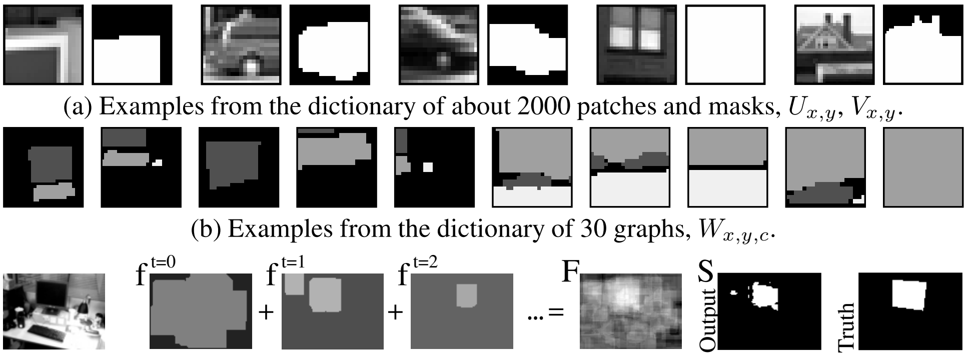
See Fig. 2 for the pseudo-code for the learning algorithm, and Fig. 3 for the pseudo-code for run-time inference. For simplicity, we have assumed that we perform a single round of boosting, $N_{boost} = 1$, and a single round of belief propagation, $N_{BP} = 1$, at each iteration.

4 BRFs for multiclass object detection and segmentation

With the BRF training algorithm in hand, we describe our approach for multiclass object detection and region-labeling using densely connected BRFs.

4.1 Weak learners for detecting stuff and things

The square sliding window approach does not provide a natural way of working with irregular objects. Using region labeling as an image representation allows dealing with irregular



(a) Examples from the dictionary of about 2000 patches and masks, $U_{x,y}$, $V_{x,y}$.

(b) Examples from the dictionary of 30 graphs, $W_{x,y,c}$.

(c) Example feedforward segmentation for screens.

Figure 4: Examples of patches from the dictionary and an example of the segmentation obtained using boosting trained with patches from (a).

and extended objects (buildings, bookshelf, road, ...). Extended stuff [1] may be a very important source of contextual information for other objects.

The weak learners we use for the local evidence potentials are based on the segmentation fragments proposed in [2]. Specifically, we create a dictionary of about 2000 image patches U, chosen at random (but overlapping each object), plus a corresponding set of binary (in-class/ out-of-class) image masks, V: see Fig. 4(a). At each round t, for each class c, and for each dictionary entry d, we construct the following weak learner, whose output is a binary matrix of the same size as the image I:

$$v^d(I) = \left((I \otimes U^d) > \theta^d\right) * V^d > 0 \tag{10}$$

where $\otimes$ represents normalized cross-correlation and $*$ represents convolution. The intuition behind this is that $I \otimes U^d$ will produce peaks at image locations that contain this patch/ template, and then convolving with V^d will superimpose the segmentation mask on top of the peaks. As a function of the threshold θ^d, the feature will behave more as a template detector ($\theta^d \simeq 1$) or as a texture descriptor ($\theta^d << 1$).

To be able to detect objects at multiple scales, we first downsample the image to scale σ, compute $w^d(I \downarrow \sigma)$, and then upsample the result. The final weak learner does this for multiple scales, ORs all the results together, and then takes a linear transformation.

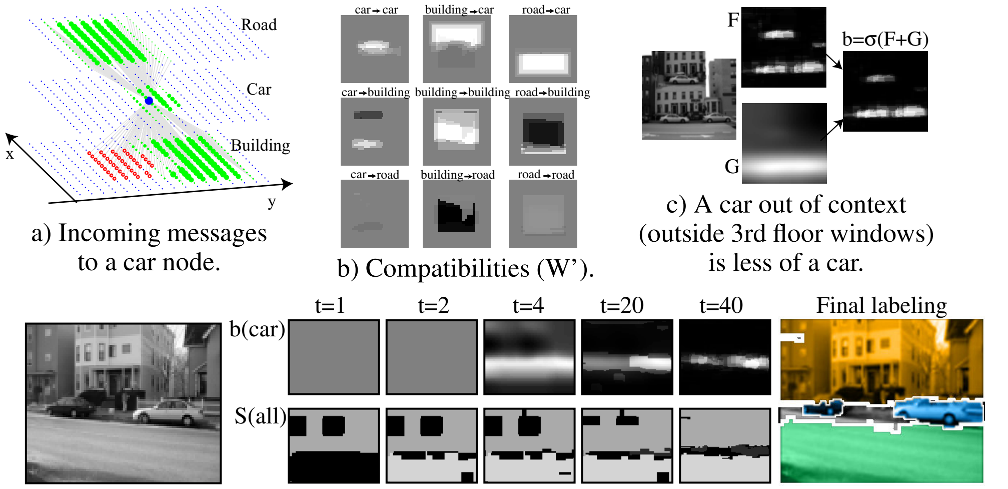
$$f^d_{x,y,c}(I) = \alpha \left(\vee_\sigma [v^d_{x,y}(I \downarrow \sigma) \uparrow \sigma]\right) + \beta \tag{11}$$

Fig. 4(c) shows an example of segmentation obtained by using boosting without context. The weak learners we use for the compatibility functions have a similar form:

$$g^d_{x,y,c}(b) = \alpha^d \left(\sum_{c'=1}^{C} b_{x',y',c'} * W^d_{x',y',c'} \right) + \beta^d \tag{12}$$

The binary kernels (graph fragments) $W^d_{x',y',c'}$ define, for each node x, y of object class c, all the nodes from which it will receive messages. These kernels are chosen by sampling patches of various sizes from the labeling of images from the training set. This allows generating complicated patterns of connectivity that reflect the statistics of object co-occurrences in the training set. The overall incoming message is given by

$$G^t_{x,y,c}(b) = \sum_{c'=1}^{C} b_{x',y',c'} * \left(\sum_n \alpha^n W^n_{x',y',c'} \right) + \sum_n \beta^n \stackrel{\text{def}}{=} \sum_{c'=1}^{C} b_{x',y',c'} * W'_{x',y',c'} + \beta'$$



a) Incoming messages to a car node.

b) Compatibilities (W').

c) A car out of context (outside 3rd floor windows) is less of a car.

d) Evolution of the beliefs for the car nodes ($b(car)$) and labeling (S) for (road, building, car).

Figure 5: Street scene. The BRF is trained to detect cars, buildings and the road.

This is computationally efficient as the function G is obtained by a convolution between the beliefs of each class and the kernel W'. We do not need to evaluate the individual g^t. Fig. 5(a-b) shows the structures of the graph and the weights $W'_{x',y',c'}$ for a BRF trained to detect cars, buildings and roads in street scenes.

4.2 Learning and inference

For training we used a labeled dataset of office and street scenes with about 100 images in each set. During the training, in the first 5 rounds we only update the local potentials, to allow local evidence to accrue. After the 5th iteration we start updating also the compatibility functions. At each round, we update only the local potential and compatibility function associated with a single object class that reduces the most the multiclass cost. This allows objects that need many features to have more complicated local potentials.

The algorithm learns to first detect easy (and large) objects, since these reduce the error of all classes the fastest. The easy-to-detect objects can then pass information to the harder ones. For instance, in office scenes, the system first detects screens, then keyboards, and finally computer mice. Fig. 6 illustrates this behavior on the test set. A similar behavior is obtained for the car detector (Fig. 5(d)). The detection of building and road provides strong constraints for the locations of the car.

4.3 Cascade of classifiers with BRFs

The BRF can be turned into a cascade [11] by thresholding the beliefs. Computations can then be reduced by doing the convolutions (required for computing f and g) only in pixels that are still candidates for the presence of the target. At each round we update a binary rejection mask for each object class, $R^t_{x,y,c}$, by thresholding the beliefs at round t: $R^t_{x,y,c} = R^{t-1}_{x,y,c}\delta(b^t_{x,y,c} > \theta^t_c)$. A pixel in the rejection mask is set to zero when we can decide that the object is not present (when $b^t_{x,y,c}$ is below the threshold $\theta^t_c \simeq 0$), and it is set to 1 when more processing is required. The threshold θ^t_c is chosen so that the percentage of missed detections is below a predefined level (we use 1%). Similarity we can define a detection mask that will indicate pixels in which we decide the object is present. The mask

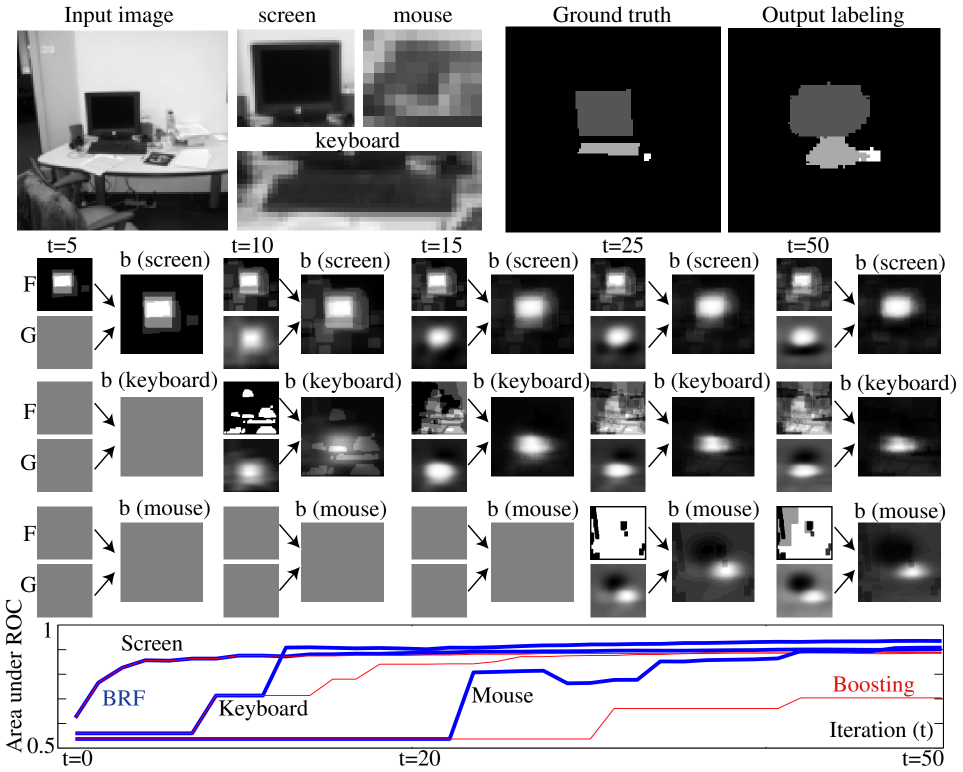


Figure 6: **Top**. In this desk scene, it is easy to identify objects like the screen, keyboard and mouse, even though the local information is sometimes insufficient. **Middle**: the evolution of the beliefs (b and F and G) during detection for a test image. **Bottom**. The graph bellow shows the average evolution of the area under the ROC for the three objects on 120 test images.

is then used for computing the features $v(I)$ and messages G by applying the convolutions only on the pixels not yet classified. We can denote those operators as $\otimes_R$ and $*_R$. This results in a more efficient classifier with only a slight decrease of performance. In Fig. 7 we compare the reduction of the search space when implementing a cascade using independent boosting (which reduces to Viola and Jones [11]), and when using BRF's. We see that for objects for which context is the main source of information, like the mouse, the reduction in search space is much more dramatic using BRFs than using boosting alone.

5 Conclusion

The proposed BRF algorithm combines boosting and CRF's, providing an algorithm that is easy for both training and inference. We have demonstrated object detection in cluttered scenes by exploiting contextual relationships between objects. The BRF algorithm is computationally efficient and provides a natural extension of the cascade of classifiers by integrating evidence from other objects in order to quickly reject certain image regions. The BRF's densely connected graphs, which efficiently collect information over large image regions, provide an alternative framework to grids for vision problems.

References

[1] E. H. Adelson. On seeing stuff: the perception of materials by humans and machines. In *Proc. SPIE*, volume 4299, pages 1–12, 2001.

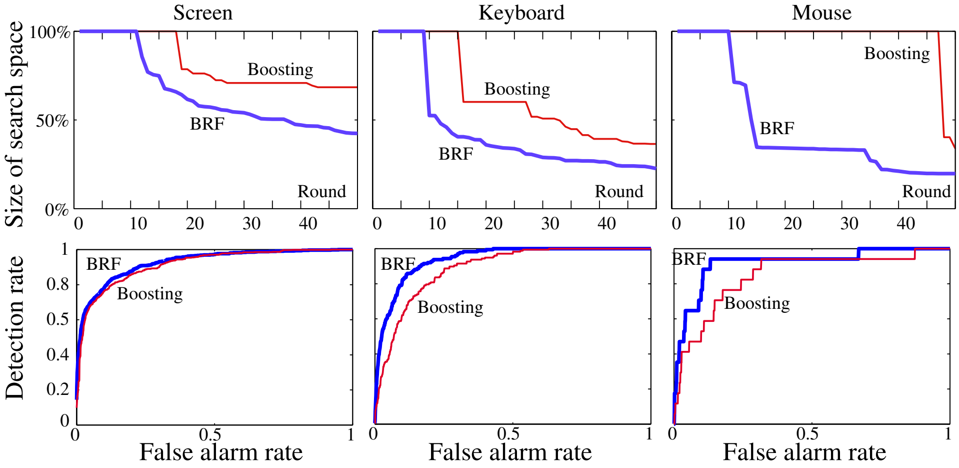


Figure 7: Contextual information reduces the search space in the framework of a cascade and improves performances. The search space is defined as the percentage of pixels that require further processing before a decision can be reached at each round. BRF's provide better performance and requires fewer computations. The graphs (search space and ROCs) correspond to the average results on a test set of 120 images.

[2] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *Proc. European Conf. on Computer Vision*, 2002.

[3] M. Fink and P. Perona. Mutual boosting for contextual influence. In *Advances in Neural Info. Proc. Systems*, 2003.

[4] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of statistics*, 28(2):337–374, 2000.

[5] Xuming He, Richard Zemel, and Miguel Carreira-Perpinan. Multiscale conditional random fields for image labelling. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004.

[6] S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

[7] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Intl. Conf. on Machine Learning*, 2001.

[8] K. Murphy, A. Torralba, and W. Freeman. Using the forest to see the trees: a graphical model relating features, objects and scenes. In *Advances in Neural Info. Proc. Systems*, 2003.

[9] R. Schapire. The boosting approach to machine learning: An overview. In *MSRI Workshop on Nonlinear Estimation and Classification*, 2001.

[10] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *Advances in Neural Info. Proc. Systems*, 2003.

[11] P. Viola and M. Jones. Robust real-time object detection. *Intl. J. Computer Vision*, 57(2):137–154, 2004.